Introduction to Big Data Assignment-1

Name: Faizan Mulla
Roll No: 21F1003885

Problem Statement

Write a Spark code to analyze user click patterns throughout the day by categorizing clicks into different time intervals (0-6, 6-12, 12-18, and 18-24 hours) using data stored in Google Cloud Storage.

Solution

I approached the problem by implementing a two-part solution using Google Cloud Platform (GCP). The solution consists of:

1. Data generator script to create csv file
2. Spark analysis script to process the data and generate required solution

Environment Setup

1. GCP Setup
   - Set up a new project with default service account configurations
   - Enabled necessary APIs (Compute Engine and Dataproc)

2. Dataproc Cluster Creation
   - Created a Dataproc cluster with following specifications:
      - chose the option: Create cluster on compute engine
      - manager node: series → e2 // machine type → e2-standard-2 (2vCPU, 1 core, 8GB memory)
      - reduce primary disk size from 500GB to something less like 50GB.
      - exact same settings for worker nodes too.

- Region: **asia-south-1**
- in the customize cluster menu, **uncheck** the INTERNAL IP ONLY option.

3. Cloud Storage Setup
   - Created a Cloud Storage bucket for storing input and output files
   - Configured with standard storage class
   - Region: **asia-south-1**

Implementation Details

1. Data Generation Script

```
import csv
import random
from datetime import timedelta, datetime


def generate_random_timestamp(base_date=datetime(2024, 10, 15)):
    return base_date + timedelta(
        days=random.randint(0, 365),
        hours=random.randint(0, 23),
        minutes=random.randint(0, 59),
        seconds=random.randint(0, 59),
    )


data = {f"user_{i + 1}": generate_random_timestamp() for i in range(0, 50)}

output_file = "data.csv"

with open(output_file, "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["id", "timestamp", "date"])
    writer.writerows(
        (user_id, ts.strftime("%Y-%m-%d %H:%M:%S"), ts.strftime("%Y-%m-%d"))
        for user_id, ts in data.items()
    )
```

2. Spark Analysis Script

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, hour, when

spark = SparkSession.builder.appName("user_click_counter").getOrCreate()
input_file = "gs://ibd-ga3/data.csv"
output_file = "output.txt"

data = spark.read.option("header", "true").option("delimiter", ",").csv(input_file)

data = data.withColumn("Hour", hour(data["timestamp"]))

data = data.withColumn(
    "time_interval",
    when(col("Hour") < 6, "00-06")
    .when(col("Hour") < 12, "06-12")
    .when(col("Hour") < 18, "12-18")
    .when(col("Hour") < 24, "18-24")
    .otherwise("Invalid timestamp"),
)

data = data.sort("time_interval")
result = data.groupBy("time_interval").count().sort("time_interval")
result.show()
result.toPandas().to_csv(output_file, sep="\t", index=False, header=True)

spark.stop()
```

Execution Process

1. Data Generation
   - Uploaded ***data-generator.py*** to Cloud Storage
   - Executed script to create sample dataset
   - Verified ***data.csv*** in Cloud Storage

2. Spark Analysis
   - Uploaded ***spark.py*** to Cloud Storage
   - Submitted Spark job through Dataproc
   - Monitored job execution through Dataproc UI

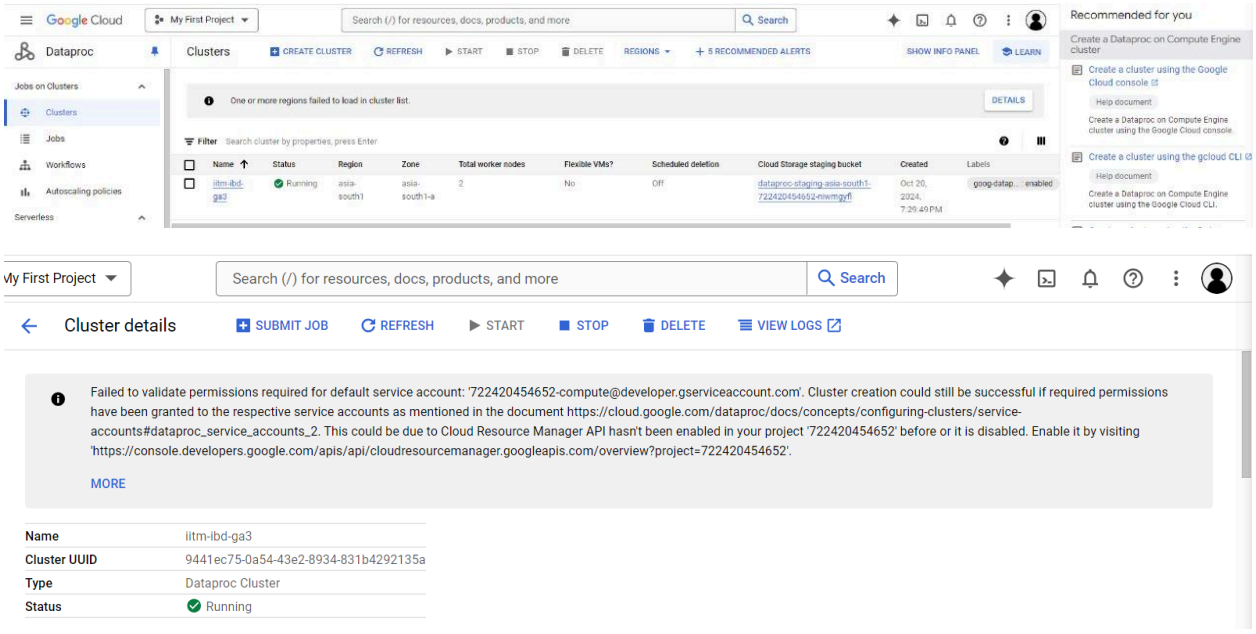
Results

The analysis successfully categorized user clicks into four time intervals:

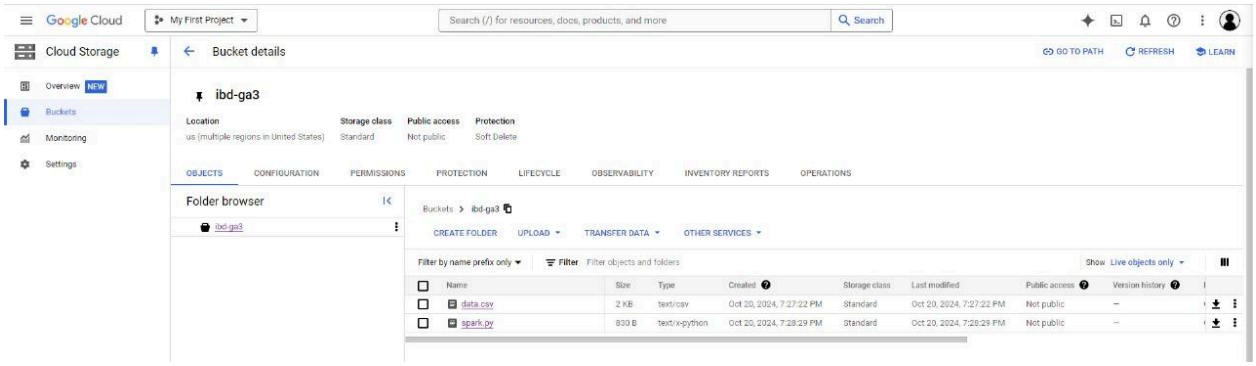
- 00-06: **14** clicks
- 06-12: **13** clicks
- 12-18: **10** clicks
- 18-24: **13** clicks

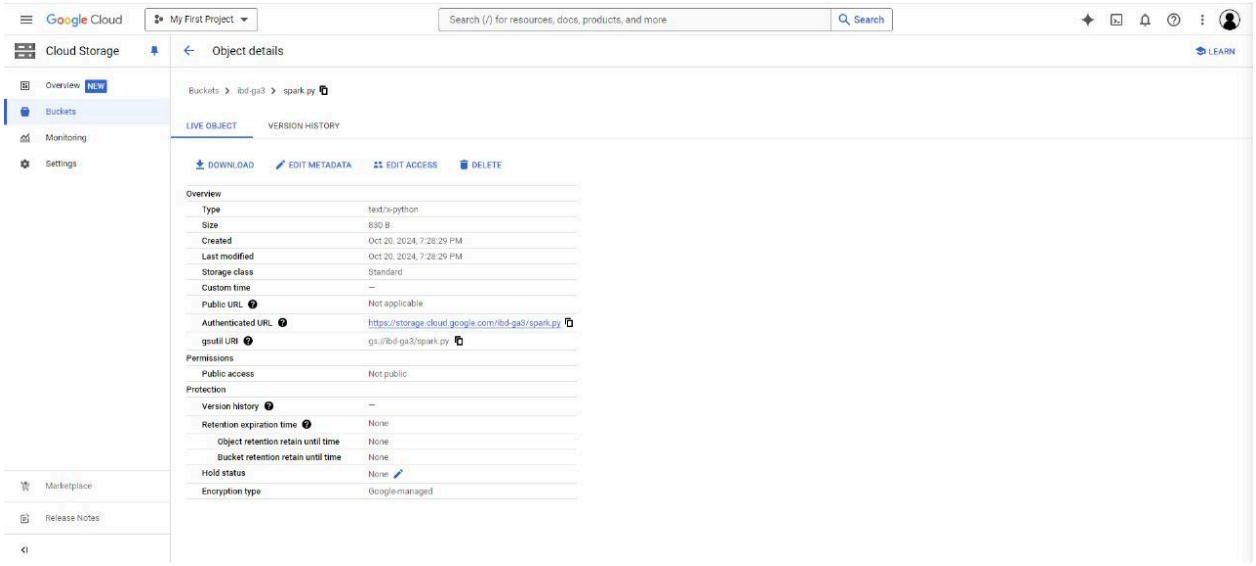
Relevant Screenshots

1. **Dataproc Cluster Configuration**

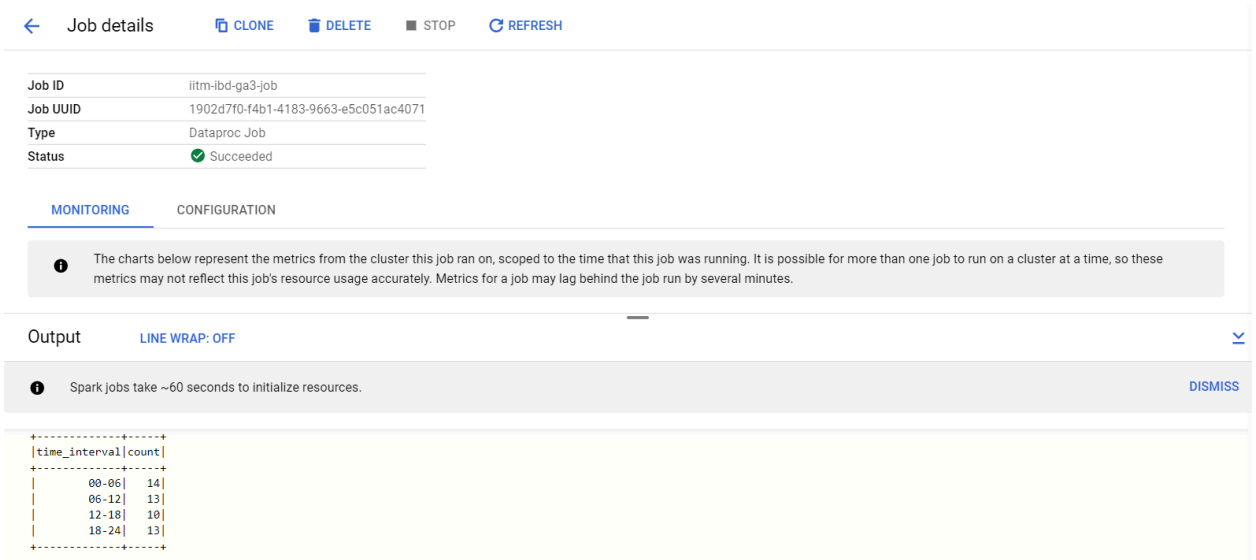


2. **Cloud Storage Bucket Contents**





3. **Job Execution Results**



4. **Output File Contents**